Chapter 5. Data Product Labels

PDS data product labels are required for describing the contents and format of each individual data product within a data set. PDS data product labels are written in the Object Description Language (ODL). The PDS has chosen to label the wide variety of data products under archival preparation by implementing a standard set of data object definitions, data elements, and standard values for the elements. These data object definitions, data elements, and standard values are defined in the *Planetary Science Data Dictionary* (PSDD). Appendix A of this document provides general descriptions and examples of the use of these data object definitions and data elements for labeling data products.

5.1 Format of PDS Labels

5.1.1 Labeling methods

In order to identify and describe the organization, content, and format of each data product, PDS requires a distinct data product label for each individual data product file. These distinct product labels may be constructed in one of three ways:

Attached - The PDS data product label is attached at the beginning of the data product file. There is one label attached to each data product file.

Detached - The PDS data product label is detached from the data and resides in a separate file which contains a pointer to the data product file. There is one detached label file for every data product file. The label file should have the same base name as its associated data file, but the extension .LBL .

Combined Detached - A single PDS detached data product label file is used to describe the contents of more than one data product file. The combined detached label contains pointers to individual data products.

NOTE: Although all three labeling methods are equally acceptable, the PDS tools do not currently support the Combined Detached label option.

Figure 5.1 illustrates the use of each of these methods for labeling individual data product files.

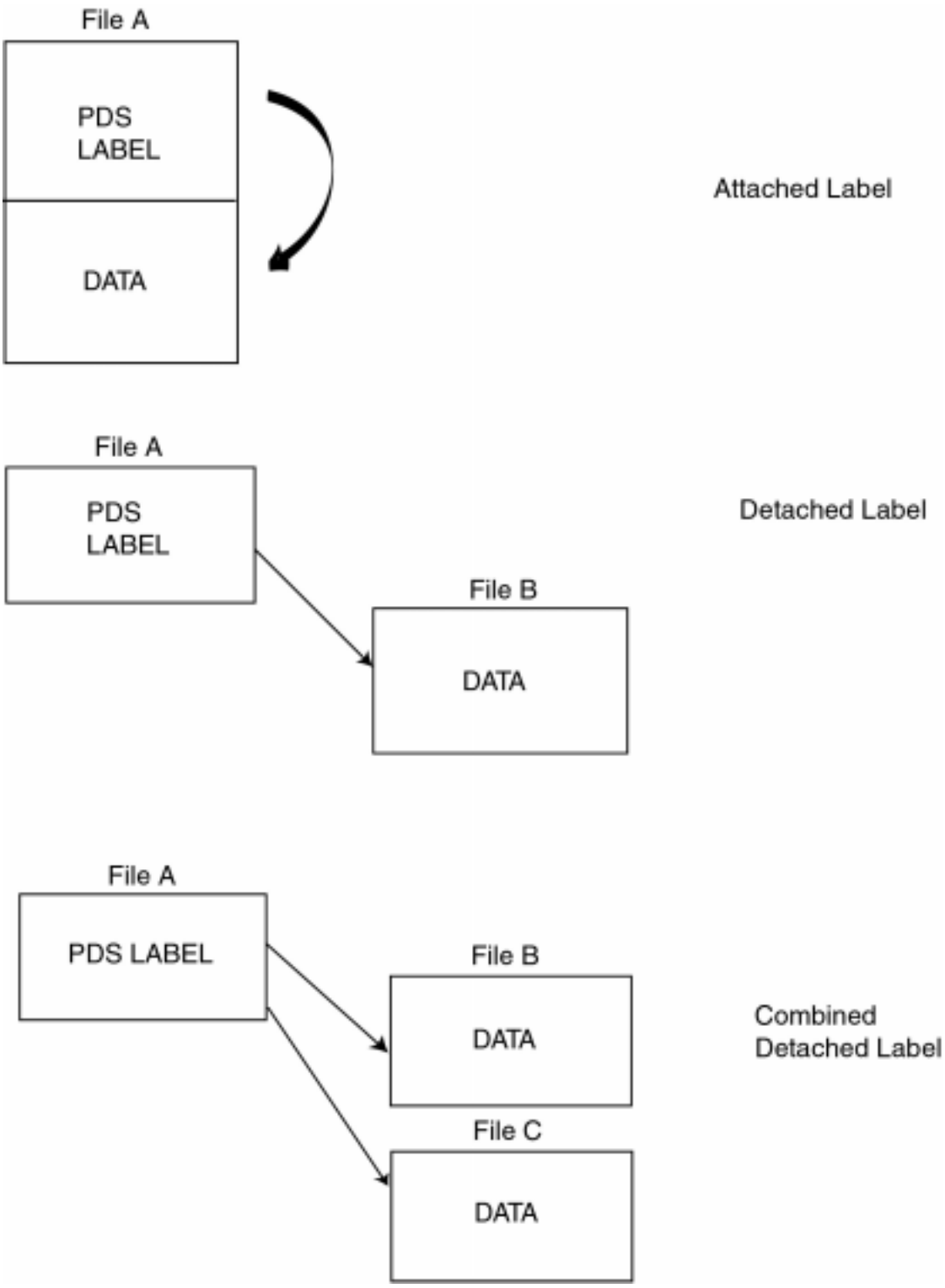

Figure 5.1 Attached, Detached, and Combined Detached PDS Labels

5.1.2 Label format

PDS recommends that labels have stream record format, and line lengths of at most 80 characters (including the CR/LF line terminators) so that the entire label can be seen on a computer screen without horizontal scrolling. The carriage return and line feed (CR/LF) pair is the required line terminator for all PDS labels. (See the *Record Formats* chapter of this document.)

All values in a PDS label should be in upper case, except values for descriptive elements (DESCRIPTION, NOTE, etc.). It is also recommended that the equal signs in the labels be aligned for ease of reading.

ASCII Character Set

All values in a PDS label must conform to the standard 7-bit ASCII character set. Labels may include characters in the range of ASCII characters 32 through 127 (decimal), and the record delimiters Line Feed (10 decimal) and Carriage Return (13 decimal).

The remaining 7-bit ASCII characters (1-9, 11, 12, and 14-31 decimal, which includes the horizontal and vertical tab and form feed characters) are not permitted in PDS labels. Note that the 8-bit characters 128 through 255 (decimal) are not used in the PDS as the interpretation of these characters varies by operating system, computer platform, and font selected. Specifically, extended-set characters with diacritical marks are not to be used as they are interpreted differently by different applications.

Label Padding

When a fixed length data file has an attached label, the label is padded with space characters (ASCII 32 decimal) in one of the following ways:

1) Spaces are added after the label's END <CR><LF> statement and before the data so that the total of the label (in bytes) is an integral multiple of the record length of the data. In this case, LABEL_RECORDS is calculated by dividing the total padded length of the label section, in bytes, by the stated value of RECORD_BYTES.

Example

In the example below, the label portion of the file is 7 x 324 = 2268 bytes in length, including blank fill between the END<CR><LF> statement and the first byte of data. The actual data portion of the file starts at record 8 (i.e., the 1st byte of the 8th record starts at byte (7 x 324)+1 = 2269)

```
RECORD_TYPE                    = FIXED_LENGTH<CR><LF>
RECORD_BYTES                   = 324<CR><LF>
FILE_RECORDS                   = 334<CR><LF>
LABEL_RECORDS                  = 7<CR><LF>

^IMAGE                         = 8<CR><LF>

END<CR><LF>
....blank fill....
data
```

2) Each line in the label may be padded with space characters so that each line in the label has the same record length as the data file. In this case, the label line length may exceed the recommended 80 characters; LABEL_RECORDS is the number of physical records in the label section of the file.

Example

In the example below, the label portion of the file is 80 x 85 = 6800 bytes in length. Each line in the label portion of the file is 85 bytes long, the same length as each data record. Notice the blank space between the actual values in the label and the line delimiters. In the example, the label is 80 lines long (i.e., 80 records long) and the data begins at record 81. Note that the label is padded so that <CR><LF> are in bytes 84 and 85.

```
RECORD_TYPE             = FIXED_LENGTH    <CR><LF>
RECORD_BYTES            = 85              <CR><LF>
FILE_RECORDS            = 300             <CR><LF>
LABEL_RECORDS           = 80              <CR><LF>
...
^TABLE                  = 81              <CR><LF>
END                                       <CR><LF>
Data
```

5.2 Data Product Label Content

5.2.1 Attached and Detached Labels

PDS data product labels have a general structure that is used for all attached and detached labels, except for data products described by minimal labels. (Minimal labels are described in Section 5.2.3.)

- LABEL STANDARDS identifier
- FILE CHARACTERISTIC data elements
- DATA OBJECT pointers
- IDENTIFICATION data elements
- DESCRIPTIVE data elements
- DATA OBJECT DEFINITIONS
- END statement

Figure 5.2 provides an example of how this general structure appears in an attached or detached label for a data product file containing multiple data objects.

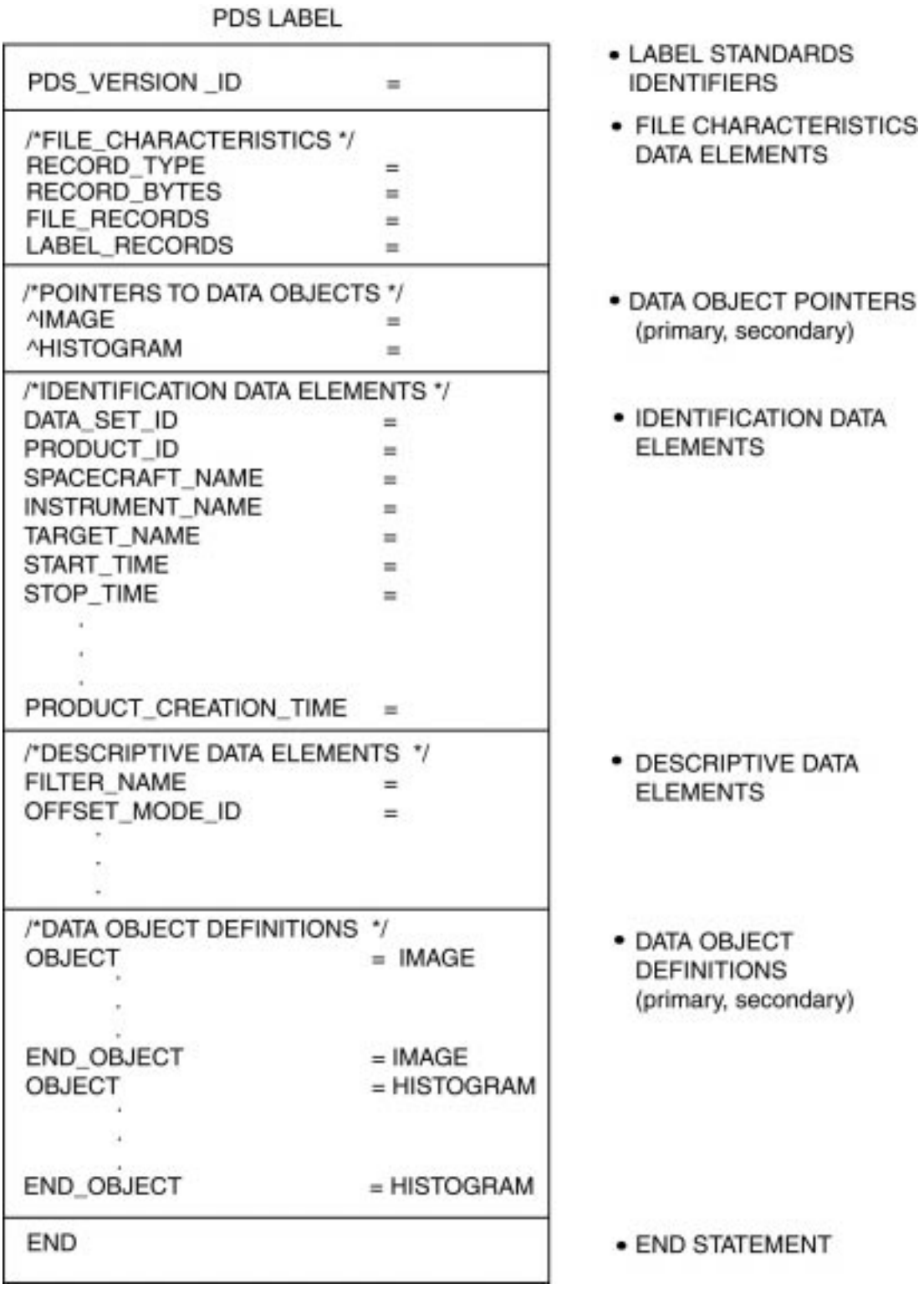

Figure 5.2 PDS Attached / Detached Label Structure

5.2.2 Combined Detached Labels

For the Combined Detached label option, the general label structure is modified slightly to explicitly reference each individual file within its own FILE object. In addition, identification and descriptive data elements that apply to all of the files can be located before the FILE objects.

- LABEL STANDARDS identifiers
- IDENTIFICATION data elements that apply to all referenced data files
- DESCRIPTIVE data elements that apply to all referenced data files
- OBJECT=FILE statement (Repeats for each data product file)
    - FILE CHARACTERISTIC data elements
    - DATA OBJECT pointers
    - IDENTIFICATION data elements
    - DESCRIPTIVE data elements
    - DATA OBJECT DEFINITION
- END_OBJECT=FILE statement
- END statement

Figure 5.3 provides an example of how this general structure appears in a combined detached label that describes more than one data product file.

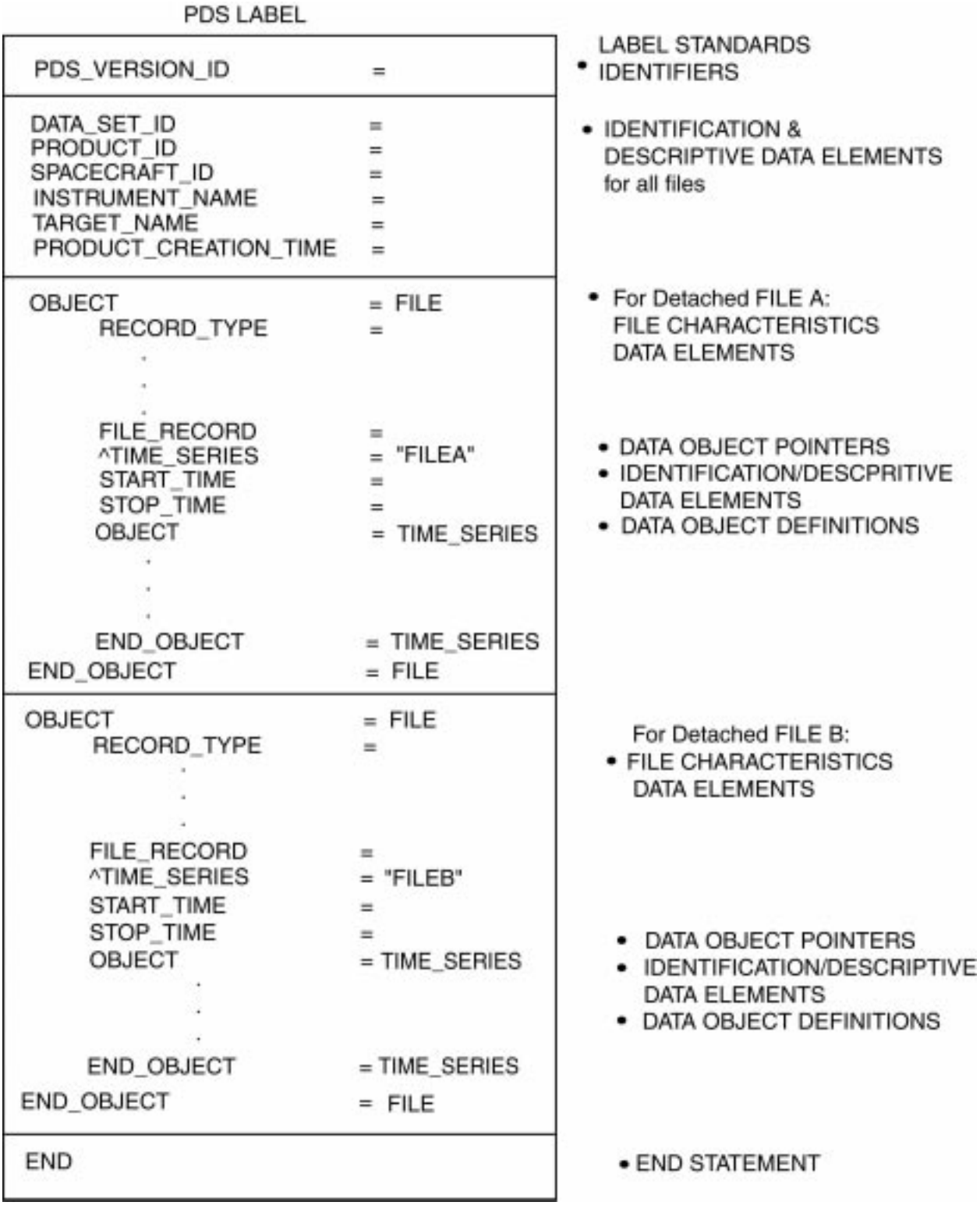

Figure 5.3 PDS Combined / Detached PDS Label Structure

5.2.3 Minimal Labels

Use of the minimal label option is only allowed when the format of the data cannot be supported by any standard PDS data object structure.

For minimal labels the required use of data objects is waived. A minimal label does not contain any PDS data object definitions or pointers to data objects. This applies to both attached and detached labels.

Minimal labels must satisfy the following requirements:

(1) Provide the ability to locate the data associated with the label.

1a. Attached labels

Since data objects and pointers are not required in the minimal label, by definition the data follows immediately after the label.

1b. Detached Labels

Both the implicit and explicit use of the FILE object are supported. The FILE_NAME keyword is required in the explicit FILE object, or in the label itself if no FILE object is included.

(2) Provide the ability to locate a description of the format/content of the data. One of the following must be provided in the minimal label:

2a. ^DESCRIPTION = “<filename>”
This is a pointer to a file containing a detailed description of the data format, which may be located in the same directory as the data or in the DOCUMENT subdirectory.

2b. DESCRIPTION = “<text appears here>”
This is either a detailed description of the data file, its format, data types, and use, or it is a reference to a document available externally, e.g., a Software Interface Specification (SIS) or similar document.

(3) When minimal labels are used, DATA_OBJECT_TYPE = FILE should be used in the DATA_SET catalog file. If a situation arises where multiple data object types are used, then separate DATA_SET catalog files should be provided for each data product. And, where appropriate, a DATA_SET_COLLECTION catalog file should also be provided.

5.2.3.1 Implicit File Object (Attached and Detached Minimal Label)

The general structure for minimal labels with implicit file objects is as follows:

- LABEL STANDARDS identifier
- FILE CHARACTERISTIC data elements
- IDENTIFICATION data elements
- DESCRIPTIVE data elements
- END statement

5.2.3.2 Explicit File Object (Detached Minimal Label)

The general structure for minimal labels with explicit file objects is as follows:

- LABEL STANDARDS identifier
- IDENTIFICATION data elements
- DESCRIPTIVE data elements
- OBJECT=FILE statement
    - FILE CHARACTERISTIC data elements

- END_OBJECT=FILE
- END statements

Figure 5.4 provides an example of how this general structure appears in a detached minimal label. In this example, an implicit FILE object is used.

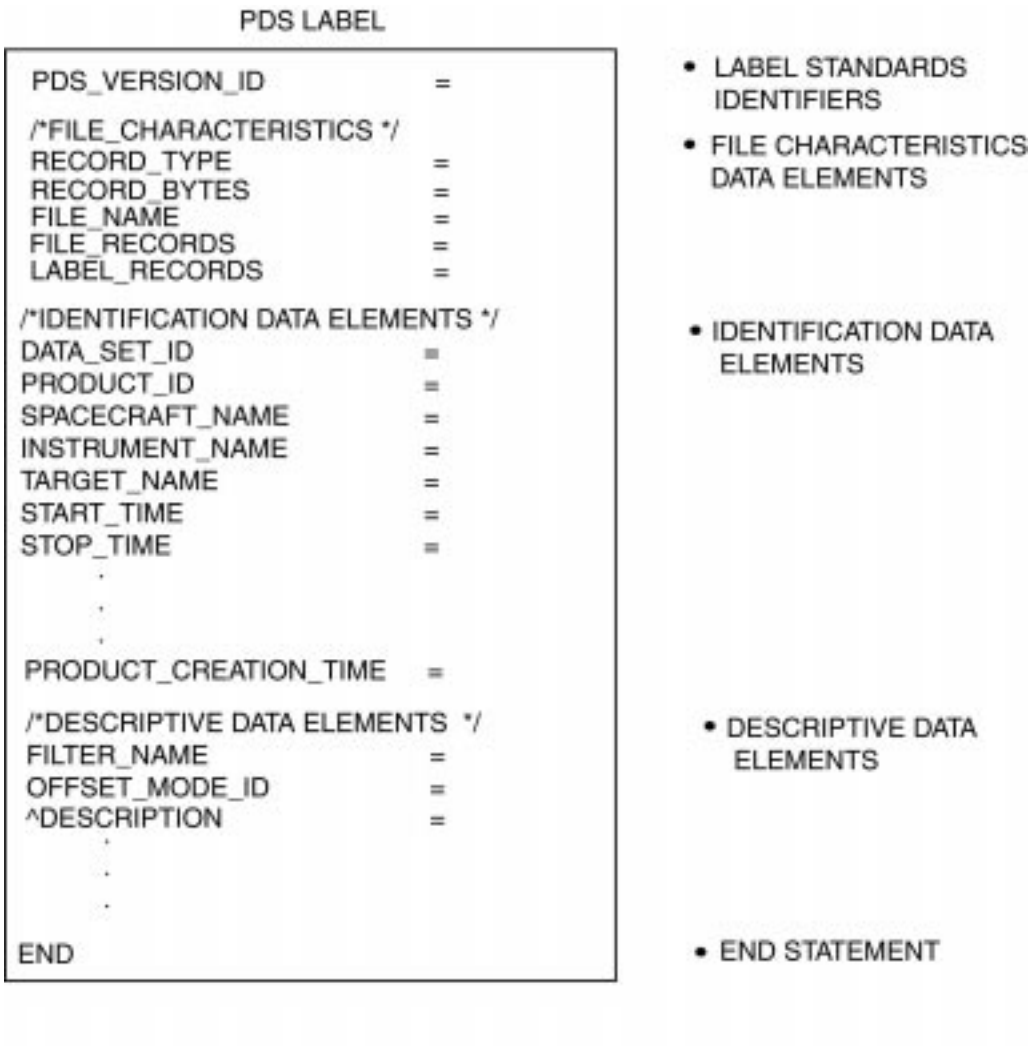

Figure 5.4 PDS Detached Minimal Label Structure

5.3 Detailed Label Contents Description

This section describes the detailed requirements for the content of PDS labels. The subsections describe label standards identifiers, file characteristic data elements, data object pointers, identification data elements, descriptive data elements, data object definitions, and the END statement.

5.3.1 Label Standards Identifiers

Each PDS label must begin with the PDS_VERSION_ID data element. This element identifies the published version of the Standards to which the label adheres, for purposes of both validation as well as software development and support. For labels adhering to the standards described in this document (the *PDS Standards Reference*, Version 3.4), the appropriate value is “PDS3”:

```
PDS_VERSION_ID = PDS3
```

The PDS does not require Standard Formatted Data Unit (SFDU) labels on individual products, but they may be desired for conformance with specific project or other agency requirements. When SFDU labels are provided on a PDS data product, the SFDU label must *precede* the PDS_VERSION_ID keyword, thus:

```
CCSD....                          [optional SFDU label]
PDS_VERSION_ID
LABEL_REVISION_NOTE
```

SFDU labels in PDS products must follow the format standards described in the *SFDU Usage* chapter in this document.

The LABEL_REVISION_NOTE element is a free form, unlimited-length character string providing information regarding the revision status and authorship of a PDS label. It should include at least the latest revision date and the author of the current version, but may include a complete editing history. This element is required in all catalog labels.

Example

```
PDS_VERSION_ID            = PDS3
LABEL_REVISION_NOTE       = "1999-08-01, Anne Raugh (SBN), initial release;"
RECORD_TYPE               = FIXED_LENGTH
RECORD_BYTES              = 80
```

5.3.2 File Characteristic Data Elements

PDS data product labels contain data element information that describes important attributes of the physical structure of a data product file. The PDS file characteristic data elements are:

RECORD_TYPE
RECORD_BYTES
FILE_RECORDS
LABEL_RECORDS

The RECORD_TYPE data element identifies the record characteristics of the data product file. A complete discussion of the RECORD_TYPE data element and its use in describing data products produced on various platforms is provided in the *Record Formats* chapter in this document. The RECORD_BYTES data element identifies the number of bytes in each physical record in the data product file. The FILE_RECORDS data element identifies the number of physical records in the file. The LABEL_RECORDS identifies the number of physical records that make up the PDS product label.

Not all of these data elements are required in every data product label. Table 5.1 lists the required (Req) and optional (Opt) file characteristic data elements for a variety of data products and labeling methods for both attached (Att) and detached (Det) labels. Where (max) is specified, the value indicates the maximum size of any physical record in the file.

Table 5.1: File Characteristic Data Element Requirements

Labeling Method	Att	Det	Att	Det	Att	Det	Att	Det
RECORD_TYPE	FIXED_LENGTH		VARIABLE_LENGTH		STREAM		UNDEFINED	
RECORD_BYTES	Req	Req	Rmax	Rmax	Omax	-	-	-
FILE_RECORDS	Req	Req	Req	Req	Opt	Opt	-	-
LABEL_RECORDS	Req	-	Req	-	Opt	-	-	-

Note: The FILE_NAME keyword is required in detached minimal labels.

5.3.3 Data Object Pointers

"Data objects" are the actual data for which the structure and attributes are defined in a PDS label. Each data product file contains one or more data objects. The PDS uses a pointer within the product labels to identify the file locations for all objects in a data product.

Example

```
^TABLE = "DATA.DAT"
^TABLE = ("DATA.DAT", 10 <BYTES>)
```

Note that, as in all non-description fields, file names must be in all uppercase characters.

5.3.3.1 Use of Pointers in Attached Labels

Data object pointers are required in labels with one exception: attached labels that refer to only a single object. In the absence of a pointer, the data object is assumed to start in the next physical record after the PDS product label area. This is commonly the case with ASCII text files described by a TEXT object and ASCII SPICE files described by a SPICE_KERNEL object. The top two illustrations in Figure 5.5 show example files that do not require data object pointers.

Object pointers are required for all data objects, even when multiple data objects are stored in a single data product file. Data object pointers in attached labels take one of two forms:

^<object_identifier> = nnn

where nnn represents the starting record number within the file (first record is numbered 1), Or,

^<object_identifier> = nnn <BYTES>

where nnn represents the starting byte location within the file (first byte is numbered 1).

See Chapter 12, *Object Description Language (ODL) Specification and Usage*, and Chapter 14, *Pointer Usage*, in this document for a complete description of pointer syntax.

The bottom two illustrations in Figure 5.5 show the use of required data object pointers for attached label products containing multiple data objects.

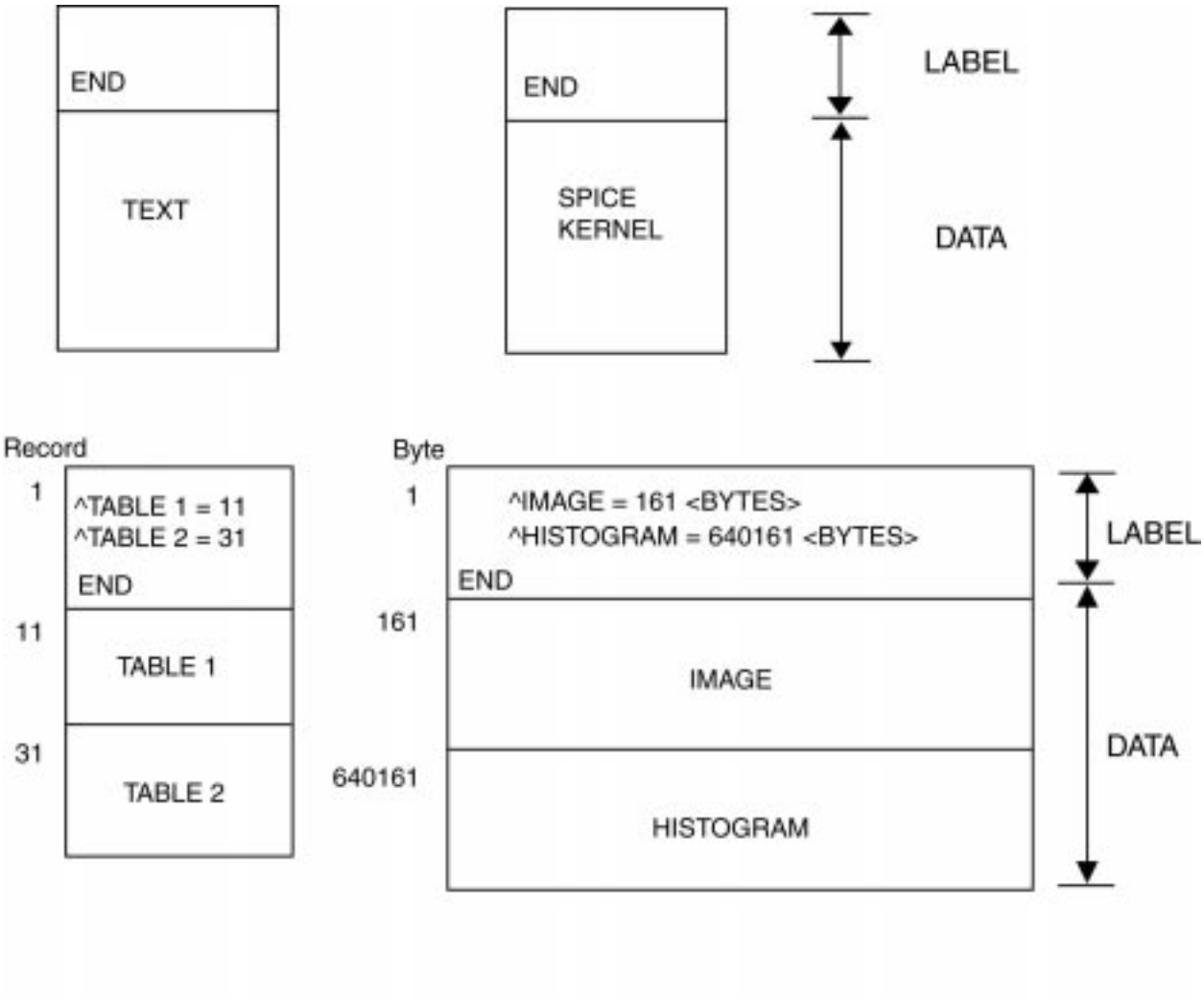

Figure 5.5 Data Object Pointers-Attached Labels

5.3.3.2 Use of Pointers in Detached and Combined Detached Labels

When the PDS data product label is a detached or a combined detached label, data object pointers are required for all data objects referenced.

The syntax for these data object pointers takes one of three forms:

(1) ^*object_identifier* = “filename”
(2) ^*object_identifier* = (“filename”, nnn)
(3) ^*object_identifier* = (“filename”, nnn <BYTES>)

With respect to the above three cases:

(a) These object pointers reference either byte or record locations in data files that are detached, or separate from, the label file.
(b) "Filename" is the name of the detached data file.
(c) When no offset is specified, the first record is assumed.
(d) Records and bytes are numbered from 1.

In the first case, the data object is located at the beginning of the referenced file. In the second case, the data object begins nnn physical records from the beginning of the referenced file. In the third case, the data object begins nnn bytes from the beginning of the referenced file.

Examples

```
^IMAGE                  = ("DATA.IMG")
^ENGINEERING_TABLE      = ("DATA.DAT", 10)
^TABLE                  = ("DATA.TAB", 10 <BYTES>)
```

Figure 5.6 contains several examples of data object pointer usage for data product files with detached or combined detached labels. The top example shows a data product consisting of a HEADER data object and a TABLE data object together in a single file. The detached label for this product includes pointers for both data objects, with the TABLE object starting at byte 601 of file A. The middle example illustrates a combined detached label for a data product contained in two data objects, each in a separate file. A separate pointer is provided for each data object. The bottom example shows a detached label for a data product containing multiple data objects.

The third example shows a complex data file structure. The HEADER object comes first in the data file and, as the pointer ("^HEADER") shows, it requires no explicit offset (record 1 is assumed). Two parallel objects, a TABLE and an IMAGE, then follow the header. For this section of the file, each record contains one row of the TABLE followed by one line of the IMAGE. In the TABLE object description, the bytes of the IMAGE are accounted for as ROW_SUFFIX_BYTES; in the IMAGE object description, the bytes of the TABLE object are accounted for as LINE_PREFIX_BYTES. Both objects start in the same record, and therefore have the same offset (4). See the IMAGE and TABLE object descriptions for more information on prefix and suffix bytes. Had this data file been organized sequentially (so that, for example, the HEADER was followed by the TABLE, which in turn was followed by the IMAGE), then each object would have had its own offset.

5.3.3.3 Note Concerning Minimal Attached and Detached Labels

Data object pointers do not exist in minimal labels. In these cases the format of the data is usually fully described in a separate file or document.

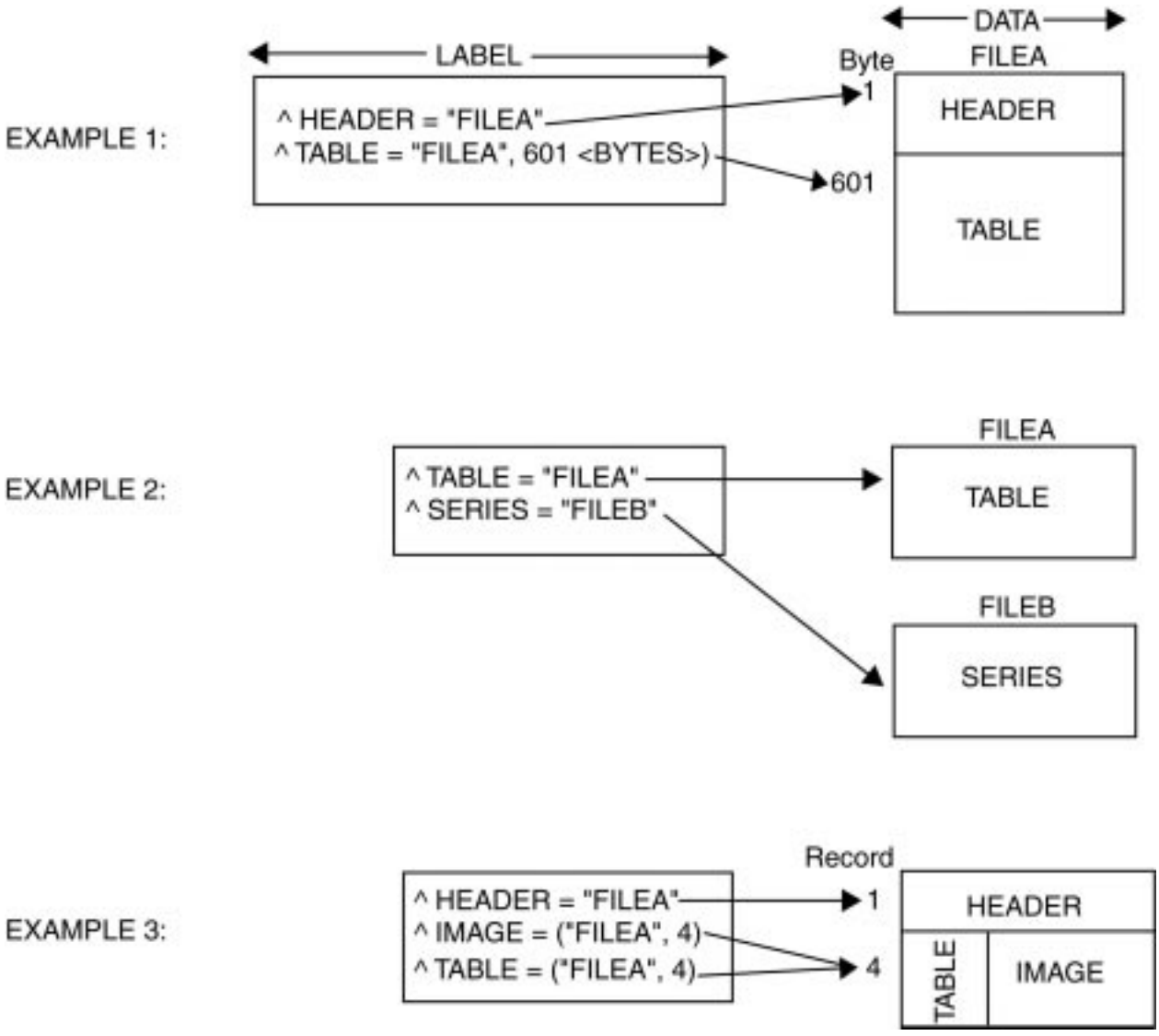

Figure 5.6 Data Object Pointers – Detached & Combined Labels

5.3.4 Data Identification Elements

The data identification elements provide additional information about a data product that can be used to relate the product to other data products from the same data set or data set collection. The minimum set of identification elements required by the PDS standards (see the following subsections) is sufficient to populate a high-level database like, for example, the PDS central catalog. In addition, data preparers will choose additional identification elements from the *Planetary Science Data Dictionary* (PSDD) to support present and future cataloging and search operations.

NOTE: When a data preparer desires a new element for a data product label - one not yet recorded in the PSDD - it can be proposed for addition to the dictionary. Contact a PDS Data Engineer for assistance.

5.3.4.1 Spacecraft Science Data Products

The following data identification elements must be included in product labels for all spacecraft science data products:

DATA_SET_ID
PRODUCT_ID
INSTRUMENT_HOST_NAME
INSTRUMENT_NAME
TARGET_NAME
START_TIME
STOP_TIME
SPACECRAFT_CLOCK_START_COUNT
SPACECRAFT_CLOCK_STOP_COUNT
PRODUCT_CREATION_TIME

5.3.4.2 Earthbased Science Data Products

The following data identification elements must be included in product labels for all Earth-based science data products:

DATA_SET_ID
PRODUCT_ID
INSTRUMENT_HOST_NAME
INSTRUMENT_NAME
TARGET_NAME
START_TIME
STOP_TIME
PRODUCT_CREATION_TIME

5.3.4.3 Ancillary Data Products

The following data identification elements must be included in product labels for all ancillary data products. Ancillary products may be more general in nature, supporting a wide variety of instruments for a particular mission. For example, SPICE data sets, general engineering data sets, and uplink data are considered ancillary data products.

DATA_SET_ID
PRODUCT_ID
PRODUCT_CREATION_TIME

The following identification elements are highly recommended, and should be included in ancillary data products whenever they apply:

INSTRUMENT_HOST_NAME
INSTRUMENT_NAME
TARGET_NAME
START_TIME
STOP_TIME
SPACECRAFT_CLOCK_START_COUNT
SPACECRAFT_CLOCK_STOP_COUNT

5.3.5 Descriptive Data Elements

In addition to the data identification elements required for various types of data, PDS strongly recommends including additional data elements related to specific types of data. These descriptive elements should include any elements needed to interpret or process the data objects or which would be needed to catalog the data product to support potential search criteria at the product level.

Recommendations for descriptive data elements to be included come from the PDS mission interface personnel as well as the data producer's own suggestions. These additional data elements are selected from the *Planetary Science Data Dictionary*.

NOTE: When a data element is needed for a data product label, but is not yet recorded in the PSDD, it may be proposed for addition to the dictionary. Contact a PDS data engineer for assistance in submitting new data elements for inclusion in the PSDD.

Pointers are sometimes used in a PDS label to provide a shorthand method for referencing either a set of descriptive data elements (e.g., ^DESCRIPTION) or a long descriptive text passage relevant to several data product labels.

5.3.6 Data Object Definitions

The PDS requires a separate data object definition within the product label for each object in the product, to describe the structure and associated attributes of each constituent object. Each object definition, whether for primary or secondary object, must have a corresponding object pointer as described in Section 5.3.3.

Object definitions are of the form:

```
OBJECT                  = aaa       where aaa is the name of the data object
...
END_OBJECT              = aaa
```

The PDS has designed a set of standard data object definitions to be used for labeling products. Among these standard objects are those designed to describe structures commonly used for scientific data storage. Appendix A provides a complete set of PDS object definition requirements, along with examples of product labels.

Pointers are sometimes used in a PDS label to provide a shorthand method for including a standard set of sub-objects referenced in several data product labels. For example, a pointer called "^STRUCTURE" is often used to include a set of COLUMN sub-objects for a TABLE structure used in many labels of the same data set.

NOTE: Minimal labels do not contain any data object definitions.

5.3.7 End Statement

The END statement ends a PDS label. Where required by an outside agency, the END statement is followed by an SFDU.

The PDS does not require SFDU labels on individual products, but they may be required to conform with specific project or other agency requirements. If SFDUs are provided on a data product, they must follow the standards described in the *SFDU Usage* chapter in this document. In some, but not all cases, another SFDU label is required after the PDS END statement to provide “end label” and sometimes “start data” information.

5.4 Syntax for Element Values

The values of keywords must be expressed in a manner appropriate to the type of the keyword. Data types for element values are specified in the element definitions contained in the PSDD. The syntax rules for expressing these values in PDS labels are discussed in detail in Section 12.3 of Chapter 12: *Object Description Language Specification and Usage*. A brief summary is provided here for reference.

Character Strings

Character strings are enclosed in double quotes unless they consist entirely of uppercase letter, number, and the underscore (_) character.

Examples

```
NAME        = FILTER                        Correct
NAME        = "FILTER WAVELENGTH"           Correct
NAME        = FILTER_WAVELENGTH             Correct

NAME        = FILTER WAVELENGTH             Incorrect
```

Integers

Integer values must be presented as a string of digits, optionally preceded by a sign. Specifically, no comma or point should be used to group digits. Values that are to be interpreted as integers must not be enclosed in quotation marks of any kind.

Examples

```
ITEMS                   = 12                Correct
REQUIRED_STORAGE_BYTES  = 43364             Correct

ITEMS                   = "12"              Incorrect
REQUIRED_STORAGE_BYTES  = 43,364            Incorrect
```

Floating-Point Numbers

Real data values may be expressed as either floating-point numbers with a decimal point or in scientific notation with an exponent. Scientific notation is formatted in the standard manner for program I/O, using the letter "E" as an exponentiation operator. Values that are to be interpreted as real numbers must not be enclosed in quotation marks of any kind.

Examples

```
TELESCOPE_LATITUDE      = 33.476                Correct
TELESCOPE_LATITUDE      = 3.3476E+01            Correct

TELESCOPE_LATITUDE      = "33.476"              Incorrect
TELESCOPE_LATITUDE      = 3.3476 x 10^01        Incorrect
```

Dates and Times

Date and time values must be in the PDS standard date/time format: *YYYY-MM-DD*T*hh:mm:ss.sss*. Date and time values must never be enclosed in quotes of any kind.

Examples

```
START_TIME  = 1990-08-01T23:59:59               Correct

START_TIME  = "1990-08-01T23:59:59"             Incorrect
```